

WEST Search History

DATE: Monday, September 19, 2005

Hide?	Set Name	Query	Hit Count
		<i>DB=PGPB; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L9	717/162,168.ccls.	373
		<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L8	714/\$.ccls. and l2	32
<input type="checkbox"/>	L7	l2 and L6	2
<input type="checkbox"/>	L6	719/328.ccls.	1017
<input type="checkbox"/>	L5	l1 and L4	175
<input type="checkbox"/>	L4	717/162,166,168.127,130,118,131.ccls.	1139
<input type="checkbox"/>	L3	L2 and except\$6	163
<input type="checkbox"/>	L2	L1 and (execut\$6 with substitu\$6)	234
<input type="checkbox"/>	L1	(load\$5 with executable\$2)	6083

END OF SEARCH HISTORY



[Subscribe](#) (Full Service) [Register](#) (Limited Service, Free) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

+abstract:load +abstract:executable +abstract:modify +abstri



Nothing Found

Your search for **+abstract:load +abstract:executable +abstract:modify +abstract:substitute** did not return any results.

You may want to try an [Advanced Search](#) for additional options.

Please review the [Quick Tips](#) below or for more information see the [Search Tips](#).

Quick Tips

- Enter your search terms in lower case with a space between the terms.

sales offices

You can also enter a full question or concept in plain language.

Where are the sales offices?

- Capitalize proper nouns to search for specific people, places, or products.

John Colter, Netscape Navigator

- Enclose a phrase in double quotes to search for that exact phrase.

"museum of natural history" "museum of modern art"

- Narrow your searches by using a + if a search term must appear on a page.

museum +art

- Exclude pages by using a - if a search term must not appear on a page.

museum -Paris

Combine these techniques to create a specific search query. The better your description of the information you want, the more relevant your results will be.

museum +"natural history" dinosaur -Chicago

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used [load](#) [executable](#) [modify](#)

Found 22 of 161,645

Sort results by

Display results

☒ [Save results to a Binder](#)
☒ [Search Tips](#)
☐ Open results in a new window

[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Results 1 - 20 of 22

 Result page: [1](#) [2](#) [next](#)

 Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Experiment management support for performance tuning](#)

Karen L. Karavanic, Barton P. Miller

 November 1997 **Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM)**

 Full text available: [pdf\(69.59 KB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The development of a high-performance parallel system or application is an evolutionary process. It may begin with models or simulations, followed by an initial implementation of the program. The code is then incrementally modified to tune its performance and continues to evolve throughout the applications's life span. At each step, the key question for developers is: *how and how much did the performance change?* This question arises comparing an implementation to models or simu ...

2 [Ease: an environment for architecture study and experimentation](#)

Jack W. Davidson, David B. Whalley

 April 1990 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1990 ACM SIGMETRICS conference on Measurement and modeling of computer systems**, Volume 18 Issue 1

 Full text available: [pdf\(220.72 KB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Gathering detailed measurements of the execution behavior of an instruction set architecture is difficult. There are two major problems that must be solved. First, for meaningful measurements to be obtained, programs that represent typical work load and instruction mixes must be used. This means that high-level language compilers for the target architecture are required. This problem is further compounded as most architectures require an optimizing compiler to exploit their capabilities. Bu ...

3 [Configurable applications for graphics employing satellites \(CAGES\)](#)

Griffith Hamlin, James D. Foley

 April 1975 **ACM SIGGRAPH Computer Graphics , Proceedings of the 2nd annual conference on Computer graphics and interactive techniques**, Volume 9 Issue 1

 Full text available: [pdf\(160.76 KB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

This paper reports on CAGES, a programming system which substantially simplifies the process of writing interactive graphics application programs for use in a distributed processing, satellite-host configuration. It allows programs written in a PL/I subset to be configurable: program modules . (main program, subroutines) and data can be easily reassigned from the host to the satellite, or vice versa. That is, the division of labor

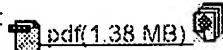
between the two computers is readily modified. The CAGES system su ...

4 Memory hierarchies: Direct load: dependence-linked dataflow resolution of load address and cache coordinate

Byung-Kwon Chung, Jinsuo Zhang, Jih-Kwon Peir, Shih-Chang Lai, Konrad Lai

December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**

Full text available:



[Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

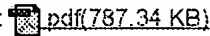
An increasing cache latency in future processors incurs profound performance impacts in spite of advanced out-of-order execution techniques. In this paper, we describe an early address resolution mechanism that accurately resolves both regular and irregular load addresses. The basic idea is to build dynamic dependence links from the instruction that updates the base register to the consumer load instructions. Once a new base address is available, it triggers calculations of the new load addresses ...

5 Exploiting Java instruction/thread level parallelism with horizontal multithreading

Kenji Watanabe, Wanming Chu, Yamin Li

January 2001 **Australian Computer Science Communications , Proceedings of the 6th Australasian conference on Computer systems architecture ACSAC '01**, Volume 23 Issue 4

Full text available:



Additional Information: [full citation](#), [abstract](#), [references](#)

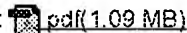
Java bytecodes can be executed with the following three methods: a Java interpreter running on a particular machine interprets bytecodes; a Just-In-Time (JIT) compiler translates bytecodes to the native primitives of the particular machine and the machine executes the translated codes; and a Java processor executes bytecodes directly. The first two methods require no special hardware support for the execution of Java bytecodes and are widely used currently. The last method requires an embedded J ...

6 Prefetching over a network: early experience with CTIP

David Rochberg, Garth Gibson

December 1997 **ACM SIGMETRICS Performance Evaluation Review**, Volume 25 Issue 3

Full text available:



Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

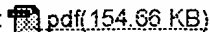
We discuss CTIP, an implementation of a network filesystem extension of the successful TIP informed prefetching and cache management system. Using a modified version of TIP in NFS client machines (and unmodified NFS servers). CTIP takes advantage of application-supplied hints that disclose the application's future read accesses. CTIP uses these hints to aggressively prefetch file data from an NFS file server and to make better local cache replacement decisions. This prefetching hides disk latency ...

7 Low power memory: Snug set-associative caches: reducing leakage power while improving performance

Jia-Jhe Li, Yuan-Shin Hwang

August 2005 **Proceedings of the 2005 international symposium on Low power electronics and design ISLPED '05**

Full text available:



Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


As transistors keep shrinking and on-chip data caches keep growing, static power dissipation due to leakage of caches takes an increasing fraction of total power in processors. Several techniques have already been proposed to reduce leakage power by turning off unused cache lines. However, they all have to pay the price of performance degradation. This paper presents a cache architecture, the *Snug Set-Associative (SSA)* cache, that does not only cut most of static power dissipation ...

Keywords: leakage power, set-associative caches

8 Informed multi-process prefetching and caching

Andrew Tomkins, R. Hugo Patterson, Garth Gibson

June 1997 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1997 ACM SIGMETRICS international conference on Measurement and modeling of computer systems**, Volume 25 Issue 1

Full text available:  pdf(2.90 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Informed prefetching and caching based on application disclosure of future I/O accesses (hints) can dramatically reduce the execution time of I/O-intensive applications. A recent study showed that, in the context of a single hinting application, prefetching and caching algorithms should adapt to the dynamic load on the disks to obtain the best performance. In this paper, we show how to incorporate adaptivity to disk load into the TIP2 system, which uses *cost-benefit analysis* to allocate g ...

9 Maintaining Consistency and Bounding Capacity of Software Code Caches

Derek Bruening, Saman Amarasinghe

March 2005 **Proceedings of the international symposium on Code generation and optimization CGO '05**

Full text available:  pdf(253.56 KB)


Additional Information: [full citation](#), [abstract](#)

Software code caches are becoming ubiquitous, in dynamic optimizers, runtime tool platforms, dynamic translators, fast simulators and emulators, and dynamic compilers. Caching frequently executed fragments of code provides significant performance boosts, reducing the overhead of translation and emulation and meeting or exceeding native performance in dynamic optimizers. One disadvantage of caching, memory expansion, can sometimes be ignored when executing a single application. However, as optimiz ...

10 Code optimization - I: Optimization opportunities created by global data reordering

Gadi Haber, Moshe Klausner, Vadim Eisenberg, Bilha Mendelson, Maxim Gurevich

March 2003 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization CGO '03**

Full text available:  pdf(1.31 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Memory access has proven to be one of the bottlenecks in modern architectures. Improving memory locality and eliminating the amount of memory access can help release this bottleneck. We present a method for link-time profile-based optimization by reordering the global data of the program and modifying its code accordingly. The proposed optimization reorders the entire global data of the program, according to a representative execution rate of each instruction (or basic block) in the code. The da ...

11 Eliminating receive livelock in an interrupt-driven kernel

Jeffrey C. Mogul, K. K. Ramakrishnan

August 1997 **ACM Transactions on Computer Systems (TOCS)**, Volume 15 Issue 3

Full text available:  ps(552.40 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Most operating systems use interface interrupts to schedule network tasks. Interrupt-driven systems can provide low overhead and good latency at low offered load, but degrade significantly at higher arrival rates unless care is taken to prevent several pathologies. These are various forms of receive livelock, in which the system spends all of its time processing interrupts, to the exclusion of other necessary tasks. Under extreme conditions,

no packets are delivered to the user ...

Keywords: interrupt-driven kernel, livelock, polling, scheduling

12 Pthreads for dynamic and irregular parallelism

Girija J. Narlikar, Guy E. Blelloch

November 1998 **Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available:  [html\(82.60 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

High performance applications on shared memory machines have typically been written in a coarse grained style, with one heavyweight thread per processor. In comparison, programming with a large number of lightweight, parallel threads has several advantages, including simpler coding for programs with irregular and dynamic parallelism, and better adaptability to a changing number of processors. The programmer can express a new thread to execute each individual parallel task; the implementation dyn ...

Keywords: Pthreads, dynamic scheduling, irregular parallelism, lightweight threads, multithreading, space efficiency

13 Automatic resumption mechanism for program debugging

Takao Shimomura

October 1991 **ACM SIGSOFT Software Engineering Notes**, Volume 16 Issue 4

Full text available:  [pdf\(442.46 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

In program debugging, tracing control instructions that examine internal program states can be saved in a file and this file can be used to initialize the debugging environment when a program is loaded by a debugger. When a source program is modified because of bugs, however, source line numbers are also changed. It is therefore necessary to update the source line numbers in a tracing control instruction file according to the modifications in the source program. This paper proposes a solution by ...

14 Accelerators: SoC design of Ogg Vorbis decoder using embedded processor

Atsushi Kosaka, Satoshi Yamaguchi, Hiroyuki Okuhata, Takao Onoye, Isao Shirakawa

April 2004 **Proceedings of the 1st conference on Computing frontiers**


Full text available:  [pdf\(644.59 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper presents an ARM-based SoC architecture for the Ogg Vorbis audio decoder. A trivial software-based implementation incurs high computational cost and requires high operation frequency. In order to achieve realtime processing and efficient bus interface design for our target system, the load of an embedded processor is reduced through the use of specific hardware for a functional block that has higher computational complexity than other blocks of Ogg Vorbis decoding process. Based on com ...

15 Energy efficient architectures: Reducing power requirements of instruction scheduling through dynamic allocation of multiple datapath resources

Dmitry Ponomarev, Gurhan Kucuk, Kanad Ghose

December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**

Full text available:  [pdf\(1.51 MB\)](#)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)
[Publisher Site](#)

The "one-size-fits-all" philosophy used for permanently allocating datapath resources in today's superscalar CPUs to maximize performance across a wide range of applications


results in the overcommitment of resources in general. To reduce power dissipation in the datapath, the resource allocations can be dynamically adjusted based on the demands of applications. We propose a mechanism to dynamically, simultaneously and independently adjust the sizes of the issue queue (IQ), the reorder buffer (R ...

Keywords: dynamic instruction scheduling, energy-efficient datapath, power reduction, superscalar processor

16 Cost-based query scrambling for initial delays

Tolga Urhan, Michael J. Franklin, Laurent Amsaleg

June 1998 **ACM SIGMOD Record , Proceedings of the 1998 ACM SIGMOD international conference on Management of data**, Volume 27 Issue 2

Full text available:  [pdf\(1.81 MB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Remote data access from disparate sources across a wide-area network such as the Internet is problematic due to the unpredictable nature of the communications medium and the lack of knowledge about the load and potential delays at remote sites. Traditional, static, query processing approaches break down in this environment because they are unable to adapt in response to unexpected delays. Query scrambling has been proposed to address this problem. Scrambling modifies query execution plans o ...

17 A brief survey of systems providing process or object migration facilities

Mark Nuttall

October 1994 **ACM SIGOPS Operating Systems Review**, Volume 28 Issue 4

Full text available:  [pdf\(1.19 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

Migration is the movement of an active entity from one machine to another during execution. Such migration may be used for dynamic load balancing purposes with the aim of gaining increased performance from a group of processors than may be gained by schemes simply allocating processes to processors at run time. Schemes providing object migration also offer object persistence, improved fault tolerance and potentially more efficient remote object invocation (RPC). The survey covers systems providin ...

18 Generating instruction sets and microarchitectures from applications

Ing-Jer Huang, Alvin M. Despain

November 1994 **Proceedings of the 1994 IEEE/ACM international conference on Computer-aided design**

Full text available:  [pdf\(751.49 KB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The design of application-specific instruction set processor(ASIP) system includes at least three interdependent tasks: microarchitecture design, instruction set design, and instruction set mapping for the application. We present a method that unifies these three design problems with a single formulation: a modified scheduling/allocation problem with an integrated instruction formation process. Micro-operations (MOPs) representing the application are scheduled into time steps. Instructions ...

19 Efficient software-based fault isolation

Robert Wahbe, Steven Lucco, Thomas E. Anderson, Susan L. Graham

December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles**, Volume 27 Issue 5

Full text available:  [pdf\(1.49 MB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

One way to provide fault isolation among cooperating software modules is to place each in its own address space. However, for tightly-coupled modules, this solution incurs prohibitive context switch overhead. In this paper, we present a software approach to implementing fault isolation within a single address space. Our approach has two parts. First, we load the code and data for a distrusted module into its own *fault domain*, a logically separate portion of the application's address space ...

20 On testing cache-coherent shared memories

Phillip B. Gibbons, Ephraim Korach

August 1994 **Proceedings of the sixth annual ACM symposium on Parallel algorithms and architectures**

Full text available:  [pdf \(1.30 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Sequential consistency is the most-widely used correctness condition for multiprocessor memory systems. High-performance shared memory multiprocessors such as the Kendall Square KSR1, the Stanford DASH, and the MIT Alewife employ a variety of techniques to improve memory system performance while providing sequential consistency. Primary among them is the use of caches at each processor, kept coherent by protocols implemented in hardware. We study the problem of testing shared mem ...

Results 1 - 20 of 22

Result page: [1](#) [2](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)


[Home](#) | [Login](#) | [Logout](#) | [Access Information](#) | [Alerts](#) |

Welcome United States Patent and Trademark Office

Search Results

[BROWSE](#)[SEARCH](#)[IEEE XPLORE GUIDE](#)

Results for "((load <in>metadata) <and> (executable<in>metadata))<and> (modificatl..."

e-mail

Your search matched 2 of 1237766 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance** in **Descending** order.

» Search Options

[View Session History](#)

Modify Search

[New Search](#)

☐ Check to search only within this results setDisplay Format: ☒ Citation ☐ Citation & Abstract

» Key

IEEE JNL. IEEE Journal or Magazine

IEEE JNL. IEE Journal or Magazine

IEEE CNF. IEEE Conference Proceeding

IEEE CNF. IEE Conference Proceeding

IEEE STD. IEEE Standard

Select Article Information

☐ 1. AUTOLOOP: Automated Action Selection in the "Observe-Analyze-Act" L Systems

Li Yin; Uttamchandani, S.; Palmer, J.; Katz, R.; Gul Agha;
 Policies for Distributed Systems and Networks, 2005. Sixth IEEE International
 06-08 June 2005 Page(s):129 - 138
 Digital Object Identifier 10.1109/POLICY.2005.9

[AbstractPlus](#) | Full Text: [PDF](#)(232 KB) IEEE CNF☐ 2. Smiley - an Interactive tool for monitoring inter-module function calls

Goldman, N.M.;
 Program Comprehension, 2000. Proceedings. IWPC 2000. 8th International W
 10-11 June 2000 Page(s):109 - 118
 Digital Object Identifier 10.1109/WPC.2000.852485

[AbstractPlus](#) | Full Text: [PDF](#)(316 KB) IEEE CNF[Help](#) [Contact Us](#) [Privacy &](#)

© Copyright 2005 IEEE -

 indexed by